

ТЕСТИРОВАНИЕ

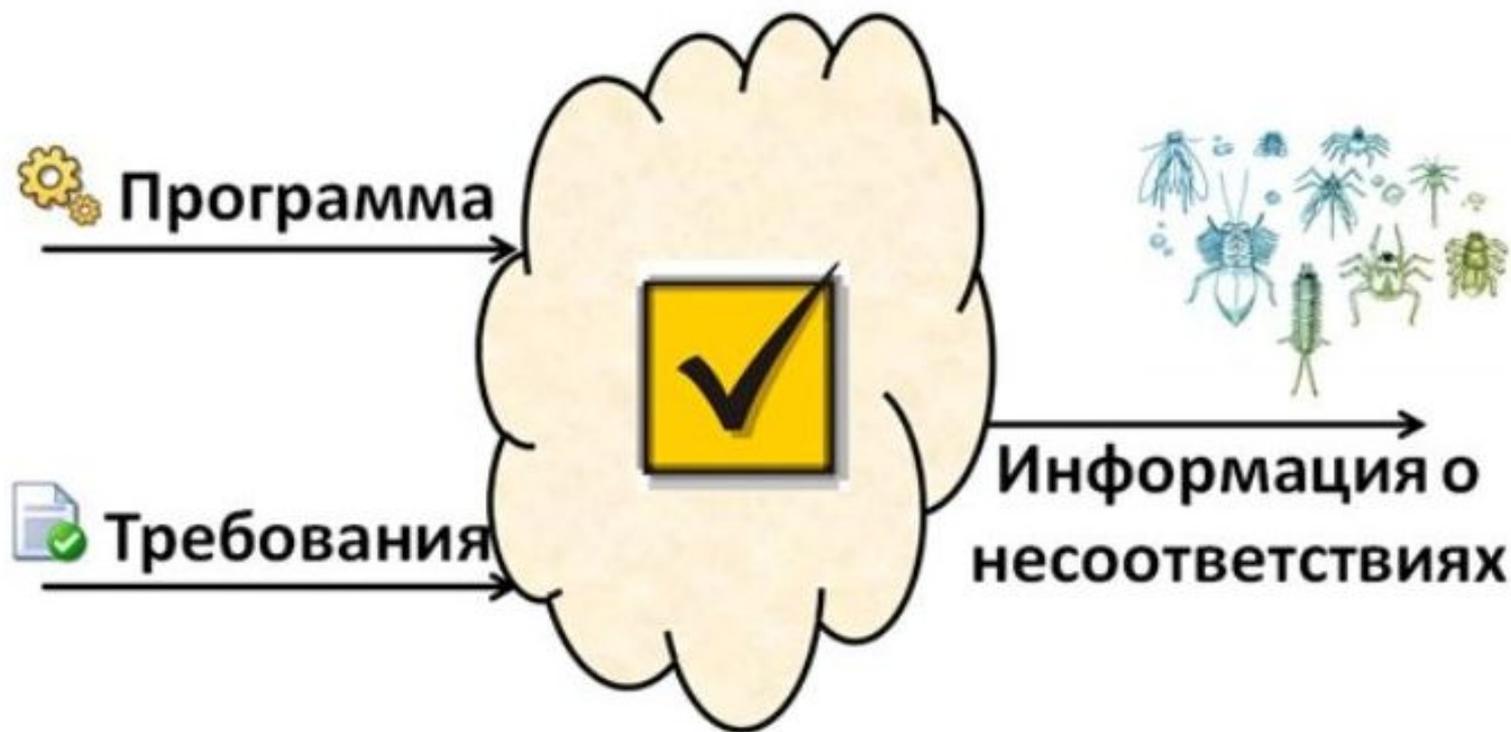
Тема: **ТЕСТИРОВАНИЕ**

ОПРЕДЕЛЕНИЕ ТЕСТИРОВАНИЯ

Тестирование программного обеспечения (Software Testing) - проверка соответствия между **реальным** и **ожидаемым** поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. [*IEEE Guide to Software Engineering Body of Knowledge, SWEBOOK, 2004*]

Тема: **ТЕСТИРОВАНИЕ**

ОПРЕДЕЛЕНИЕ ТЕСТИРОВАНИЯ



Ожидаемое поведение должно быть зафиксировано в требованиях

Тема: **ТЕСТИРОВАНИЕ**

ЧТО МОЖНО ТЕСТИРОВАТЬ?

Программы (software) при их непосредственном запуске и исполнении



Код (code) программ без запуска и исполнения.



Тема: **ТЕСТИРОВАНИЕ**

ЧТО МОЖНО ТЕСТИРОВАТЬ?

Прототип программного продукта
(product prototype)



Тема: **ТЕСТИРОВАНИЕ**

ЧТО МОЖНО ТЕСТИРОВАТЬ?

Проектная документация (project documentation):

- Требования к программному продукту (product requirements).
- Функциональные спецификации к программному продукту (functional specifications).
- Архитектура (architecture) и дизайн (design).
- План проекта (project plan) и тестовый план (test plan).
- Тестовые случаи и сценарии (test cases, test scenarios).

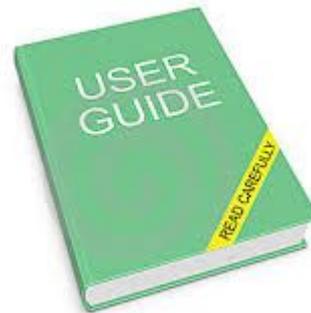
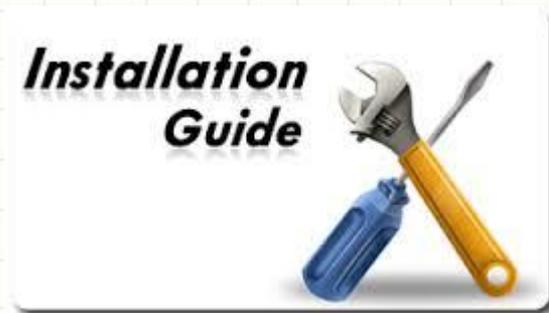


Тема: **ТЕСТИРОВАНИЕ**

ЧТО МОЖНО ТЕСТИРОВАТЬ?

Сопроводительная документация (и документация для пользователей):

- Интерактивная помощь (on-line help).
- Руководства по установке (Installation guide) и использованию программного продукта (user manual).



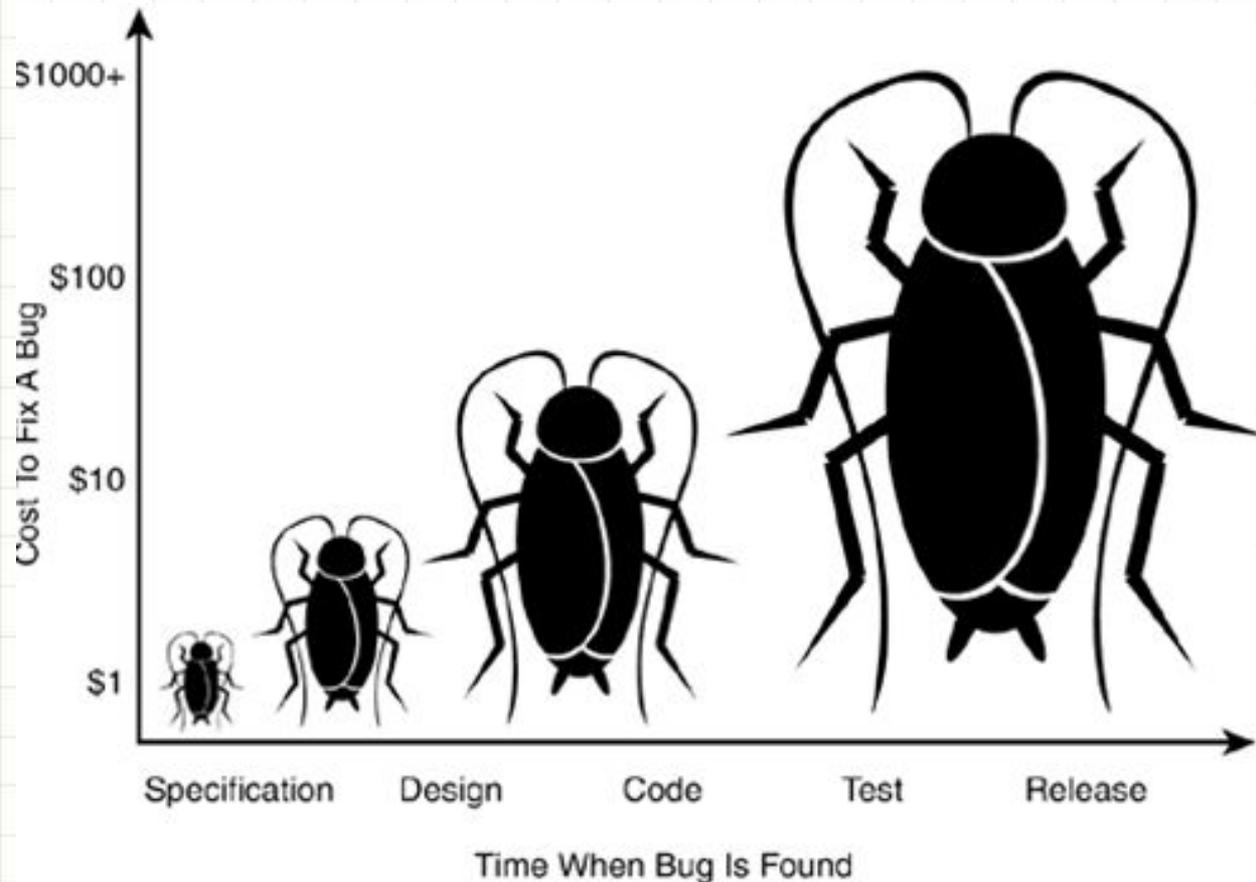
dreamstime.com



Тема: **ТЕСТИРОВАНИЕ**

ВАЖНОСТЬ ТЕСТИРОВАНИЯ

Стоимость устранения дефекта (bug) на различных стадиях проекта



Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Дефект (баг, глюк; defect, bug) – любое несоответствие фактического и ожидаемого результата (согласно требованиям или здравому смыслу).



Bug Free Software

Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Ожидаемый результат (expected result) – такое поведение программного средства, которое мы ожидаем в ответ на наши действия.



Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Тест-план (test plan) – часть проектной документации, описывающая и регламентирующая процесс тестирования.



Cash Incorporated

Project Name	Money Generator Suite
Product Name	Coin Generator
Product Release Version	9.1.8

Master Test Plan

Document Version: 1.0

Date: 01/02/2059
Prepared by: John Doe

Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Тест-план (test plan) – часть проектной документации, описывающая и регламентирующая процесс тестирования.



Cash Incorporated

Project Name	Money Generator Suite
Product Name	Coin Generator
Product Release Version	9.1.8

Master Test Plan

Document Version: 1.0

Date: 01/02/2059
Prepared by: John Doe

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ТЕСТ ПЛАНЕ

Что надо
тестировать?

Что будете
тестировать?

- перечень функциональности, которые будут подвергаться тестированию.
- перечень функциональности, которые не будут тестироваться по тем или иным причинам.

Как будете
тестировать?

описание процесса тестирования с точки зрения применяемых видов и методов тестирования

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ТЕСТ ПЛАНЕ

**Когда
тестировать?**

**Что для
тестирования
необходимо?**

**необходимое оборудование,
человеческие, финансовые ресурсы и
временные затраты на проведение
тестирования.**

**Критерии
начала
тестирования**

**Например:
готовность тестовой платформы (тестового
стенда)
законченность разработки требуемого
функционала
наличие всей необходимой документации
...**

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ТЕСТ ПЛАНЕ

Критерии
завершения
тестирования

требования к количеству открытых багов

Тема: ТЕСТИРОВАНИЕ

ПРИМЕР ТЕСТ ПЛАНА



QATestLab
154a, Dvorzhichyvska str.,
Kyiv, Ukraine
Phone: +380-44-4057024
<http://qa-testlab.com/>
contact@qa-testlab.com

Deliverables

Review

The final outcome of the testing should include the detailed result of the testing with the found defects and problems, recommendations for product improvement, and suggestions for additional testing of hotfixes.

<http://qatestlab.com/ru/knowledge-center/Sample-Deliverables/Development-of-Test-Plan/>

Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Чек-лист (check-list) –
набор идей тестов.



Тема: **ТЕСТИРОВАНИЕ**

ПРИМЕР ЧЕК-ЛИСТА ДЛЯ ТЕКСТОВОГО ПОЛЯ

Проверка вводимых символов

- Пустое поле
- Несколько пробелов
- Пробелы до и после текста
- Текст в верхнем регистре
- Текст в нижнем регистре
- Текст в верхнем и нижнем регистре
- Знаки препинания
- Цифры
- Длинные строки: 255, 256, 257, 1000, 1024, 2000, 2048 и более символов
- Спецсимволы: ~`!@#\$%^&*()_+<>?:"{}[];'
- Символы на разных языках

Тема: **ТЕСТИРОВАНИЕ**

ПРИМЕР ЧЕК-ЛИСТА ДЛЯ ТЕКСТОВОГО ПОЛЯ

Проверка способа ввода

- Сотрите несколько символов клавишей BackSpace, а потом введите другие
- Передвиньте курсор стрелками в середину слова, и впишите туда несколько символов, включая пробел
- Передвиньте курсор мышью в середину слова, и впишите туда несколько символов, включая пробел
- Вставьте несколько символов командой "Paste" или сочетанием "Ctrl-V"
- Скопируйте и вставьте несколько абзацев для проверки обработки перевода строки.
- Имитируйте перевод второй половины текста на новую строку с помощью многократного нажатия клавиши "Пробел"
- Нажмите кнопку отправки и попытайтесь в процессе отправки быстро ввести в форму ещё несколько символов

Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Тест-кейс (test case) – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

Тема: ТЕСТИРОВАНИЕ

ПРИМЕР ТЕСТ-КЕЙСА

Название:	Тест отправки сообщения		
Функция:	Контакт-Вопросы		
Действие	Ожидаемый результат	Результат теста: <ul style="list-style-type: none">• пройден• провален• заблокирован	
Предусловие:			
Откройте сайт Про Тестинг: http://www.protesting.ru	Сайт Про тестинг открыт и доступен		
Перейдите по ссылке " Задать вопрос " (внизу страницы)	Страница "Вопросы, пожелания и заявки" открыта и доступна		

Тема: ТЕСТИРОВАНИЕ

ПРИМЕР ТЕСТ-КЕЙСА

Шаги теста:		
<p>Заполните форму отправки комментария:</p> <p>"Тип Обращения": Консультация "Контактное лицо": Ольга "E-mail": test@test.com "Сообщение": <i>Добрый день, уважаемый коллектив "ПроТестинг"! Я еще ни разу не видела кач. примера баг-репорта, тест-кейса и прочей необходимой док-ии. Не подскажите, где я могу с ними ознакомиться? С уважением.</i></p>	Данные успешно введены	
Нажмите кнопку "Отправить"	Страница "Ваш запрос успешно отправлен!" открыта	

Тема: **ТЕСТИРОВАНИЕ**

ПРИМЕР ТЕСТ-КЕЙСА

Постусловие:		
Кликните по ссылке " Перейти Назад на форму отправления заявок "	Страниц "Вопросы, пожелания и заявки" открыта	

Тест кейс взят с сайта <http://www.protesting.ru/testing/templates.html>

Тема: **ТЕСТИРОВАНИЕ**

ОСНОВНЫЕ ТЕРМИНЫ

Дефект (*bug*) – выявленное несоответствие между ожидаемым (зафиксированным в требованиях) и реальным поведением программы

Отчет о дефекте (*bug report*) – это технический документ, написанный с целью:

- предоставить информацию о проблеме;
- приоритизировать проблему;

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ОТЧЕТЕ ОБ ОШИБКАХ

Идентификатор
(id)

Заголовок вида
“Что? Где? При
каких
условиях?”

Например
«Отсутствует логотип на странице
приветствия, если пользователь
является администратором».

Критичность

- **Blocked**
- **Critical**
- **Major**
- **Minor**
- **Trivial**

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ОТЧЕТЕ ОБ ОШИБКАХ

Подробное
описание
(description) с
фактическим и
ожидаемым
результатом

Требование: R245.3.23b

**Шаги
воспроизведения**

Описывает последовательность действий, которые необходимо выполнить для воспроизведения ошибки.

Тема: **ТЕСТИРОВАНИЕ**

ЧТО ДОЛЖНО БЫТЬ В ОТЧЕТЕ ОБ ОШИБКАХ

Воспроизводимость
(reproducible,
reproducibility)

Другая информация

Например:

- браузер и его версия, в котором производилось тестирование,
- Название и версия операционной системы и т.д.

Приложения
(Attachments)

Принт-скрины, видео файлы с фиксацией процесса тестирования.

Тема: **ТЕСТИРОВАНИЕ** **УРОВНИ КРИТИЧНОСТИ (Severity) ДЕФЕКТА**

- **Блокирующая (Blocker)**

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

- **Критическая (Critical)**

Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

Тема: **ТЕСТИРОВАНИЕ** **УРОВНИ КРИТИЧНОСТИ (Severity) ДЕФЕКТА**

- **Значительная (Major)**
Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.
- **Незначительная (Minor)**
Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.
- **Тривиальная (Trivial)**
Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредством пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Тема: **ТЕСТИРОВАНИЕ**

ПРИМЕР ОТЧЕТА ОБ ОШИБКЕ

Идентификатор: PASS_RZD_Refresh_001

Критичность: **Попытайтесь определить сами?**

Краткое описание: Отсутствие обновления информации в форме “Выбор маршрута” по кнопке Refresh.

Детальное описание:

Обновление в форме “Выбор маршрута” на стартовой странице по кнопке Refresh после перехода на страницу выбора поезда и возврата из нее по кнопке Back не происходит

Тема: ТЕСТИРОВАНИЕ

ПРИМЕР ОТЧЕТА ОБ ОШИБКЕ

Ожидаемый результат:

Страница обновилась. Информация в полях “Станция отправления и прибытия”, “Дата и время” соответствует

The screenshot shows a web interface for purchasing railway tickets. At the top, there are three tabs: "Купить билет и посмотреть расписание" (selected), "Билет на электричку", and "Движение поездов онлайн". Below the tabs, the search criteria are displayed:

- Станции отправления и прибытия:** "Откуда" (From) is set to "Москва, Петрозаводск, Тверь, Нижний Новгород, Вологда" and "Куда" (To) is set to "Санкт-Петербург, Казань, Ярославль, Минск, Киев".
- Дата и время отправления:** "Туда" (One-way) is set to "10.07.2015, ПТ" (Friday, July 10, 2015) with a time of "00:00-24:00". "Обратно" (Round-trip) is set to "11.07.2015, СБ" (Saturday, July 11, 2015) with a time of "00:00-24:00".

At the bottom, there are several options and links:

- Поезда
- Электрички
- [Оформить заявку на перевозку автомобиля](#) **NEW**
- [Купить премиальный билет](#) **NEW**
- [Видеоинструкция по покупке билета](#) **NEW**
- [Оферта](#)
- Только с билетами
- [Расписание](#)

Тема: ТЕСТИРОВАНИЕ

ПРИМЕР ОТЧЕТА ОБ ОШИБКЕ

Полученный результат:

Информациях в полях “Станция отправления и прибытия”, “Дата и время” не обновилась.

The screenshot shows a web interface for purchasing train tickets. The main heading is "Купить билет и посмотреть расписание" (Buy ticket and view schedule). There are three tabs: "Билет на электричку" (Ticket for commuter train), "Движение поездов онлайн" (Train movement online), and "Купить билет и посмотреть расписание" (Buy ticket and view schedule). The "Купить билет и посмотреть расписание" tab is active.

The interface displays the following information:

- Станции отправления и прибытия** (Departure and arrival stations):
 - From: МОСКВА (Moscow). Below it: Москва, Петрозаводск, Тверь, Нижний Новгород, Вологда.
 - To: САНКТ-ПЕТЕРБУРГ (Saint-Petersburg). Below it: Санкт-Петербург, Казань, Ярославль, Минск, Киев.
- Дата и время отправления** (Date and time of departure):
 - Tуда (There): 12.08.2015, СР (12.08.2015, Wednesday). Time: 7⁰⁰-18⁰⁰.
 - Обратно (Back): 11.07.2015, СБ (11.07.2015, Saturday). Time: 00⁰⁰-24⁰⁰.

At the bottom, there are several links and options:

- Поезда (Trains)
- Электрички (Commuter trains)
- [Оформить заявку на перевозку автомобиля](#) (Apply for car transport) with a "NEW" badge.
- [Купить премиальный билет](#) (Buy premium ticket) with a "NEW" badge.
- [Видеоинструкция по покупке билета](#) (Video instruction on ticket purchase) with a "NEW" badge.
- [Оферта](#) (Terms and conditions).
- Только с билетами (Only with tickets).
- [Купить билет](#) (Buy ticket) button.

Тема: **ТЕСТИРОВАНИЕ**

ПРИМЕР ОТЧЕТА ОБ ОШИБКЕ

Шаги воспроизведения:

- <http://pass.rzd.ru>
- Ввести конечную и начальную станции, дату и указать диапазон времени.
- Нажать на кнопку “Купить Билет”
- Вернуться с помощью кнопки Back на стартовую страницу.
- Нажать кнопку Refresh.

Тема: **ТЕСТИРОВАНИЕ**

ТИПЫ ТЕСТИРОВАНИЯ

Типы тестирования

По степени автоматизации

Автоматизированное

Мануальное

По критерию запуска (запускается ПО или нет)

Статическое

Динамическое

По знанию внутренней реализации продукта

Тестирование черного ящика (Black Box)

Тестирование белого ящика (White Box)

Тестирование серого ящика (Grey Box)

Тема: **ТЕСТИРОВАНИЕ**

АВТОМАТИЗИРОВАННОЕ vs. МАНУАЛЬНОЕ ТЕСТИРОВАНИЕ

- При **ручном тестировании** (мануальном) тестировщики **вручную** выполняют тесты, не используя никаких средств автоматизации.
- **Автоматизированное тестирование** предполагает использование специального программного обеспечения (помимо тестируемого) для контроля выполнения тестов и сравнения ожидаемого и фактического результата работы программы.

Тема: **ТЕСТИРОВАНИЕ**

СТАТИЧЕСКОЕ vs. ДИНАМИЧЕСКОЕ ТЕСТИРОВАНИЕ

- **Статическое тестирование** – тип тестирования, который предполагает, что программный код во время тестирования не будет выполняться. При этом само тестирование может быть как ручным, так и автоматизированным.

Например, к статическому тестированию относят вычитку кода программы и тестирование требований.

- **Динамическое тестирование** тип тестирования, который предполагает запуск программного кода. Таким образом, анализируется *поведение* программы во время ее работы.

Тема: **ТЕСТИРОВАНИЕ**

Black Box, White Box, Grey Box тестирование

- **Black Box** - тестирование, как функциональное, так и нефункциональное, не предполагающее знания внутреннего устройства компонента или системы.

Это тестирование по спецификации.

Пример: Тестировщик проводит тестирование веб-сайта, не зная особенностей его реализации, используя только предусмотренные разработчиком поля ввода и кнопки. Источник ожидаемого результата – спецификация.

Тема: **ТЕСТИРОВАНИЕ**

Black Box, White Box, Grey Box тестирование

- **White Box** - метод тестирования программного обеспечения, который предполагает, что внутренняя структура/устройство/реализация системы известны тестировщику. Мы выбираем входные значения, основываясь на знании кода, который будет их обрабатывать. Точно так же мы знаем, каким должен быть результат этой обработки.

Пример: Тестировщик, который, как правило, является программистом, изучает реализацию кода поля ввода на веб-странице, определяет все предусмотренные (как правильные, так и неправильные) и не предусмотренные пользовательские вводы, и сравнивает фактический результат выполнения программы с ожидаемым. При этом ожидаемый результат определяется именно тем, как должен работать код программы.

Тема: **ТЕСТИРОВАНИЕ**

Black Box, White Box, Grey Box тестирование

Grey Box– метод тестирования программного обеспечения при котором, внутреннее устройство программы нам известно лишь частично.

Предполагается, например, доступ к внутренней структуре и алгоритмам работы ПО для написания максимально эффективных тест-кейсов, но само тестирование проводится с помощью техники черного ящика, то есть, с позиции пользователя.

Тема: **ТЕСТИРОВАНИЕ**

Некоторые тестирования

- **Функциональное тестирование** (functional testing)
- **Тестирование взаимодействия**
- **Тестирование безопасности** (security testing)
- **Тестирование локализации** (localization testing)
- **Тестирование производительности** (performance testing)
- **Тестирование удобства использования** (usability testing)
- **Конфигурационное тестирование** (configuration testing)
- **Инсталляционное тестирование** (installation testing)
- **Тестирование на отказ и восстановление** (Failover and Recovery Testing)

Тема: **ТЕСТИРОВАНИЕ**

Виды тестирования

Функциональное тестирование (functional testing) – проверка продукта на соответствие требованиям и спецификациям.

Цель:

- **Обнаружить дефекты** в программном продукте.
- **Определить степень соответствия** программного продукта требованиям и ожиданиям заказчика.
- **Принять решение** о возможности передачи продукта заказчику.

Тема: ТЕСТИРОВАНИЕ

Виды тестирования

Тестирование взаимодействия – вид тестирования, нацеленный на оценку качества взаимодействия компонент (модулей) или всей разрабатываемой с другими компонентами или программным обеспечением.

Тема: **ТЕСТИРОВАНИЕ**

Виды тестирования

Тестирование безопасности – проверка безопасности системы и анализ рисков, связанных с атаками хакеров, вирусов, несанкционированного доступа к конфиденциальной информации.



Тема: **ТЕСТИРОВАНИЕ**

Виды тестирования

Нагрузочное тестирование или тестирование производительности - это автоматизированное тестирование, имитирующее работу определенного количества пользователей на каком-либо общем (разделяемом ими) ресурсе.

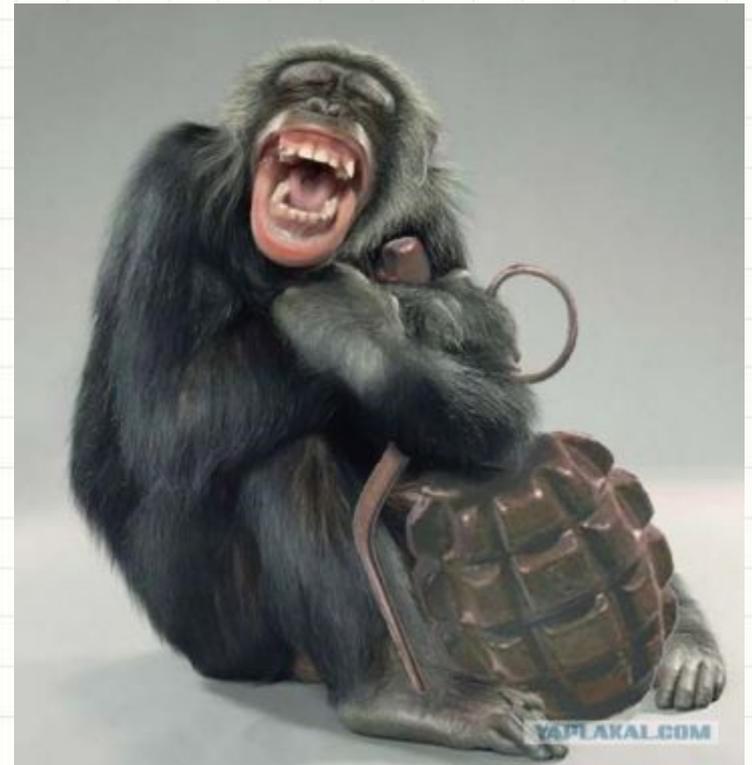


Тема: **ТЕСТИРОВАНИЕ**

Тестирование удобства использования

Метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта.

ПОНЯТНО И ПРИЯТНО!



Тема: **ТЕСТИРОВАНИЕ**

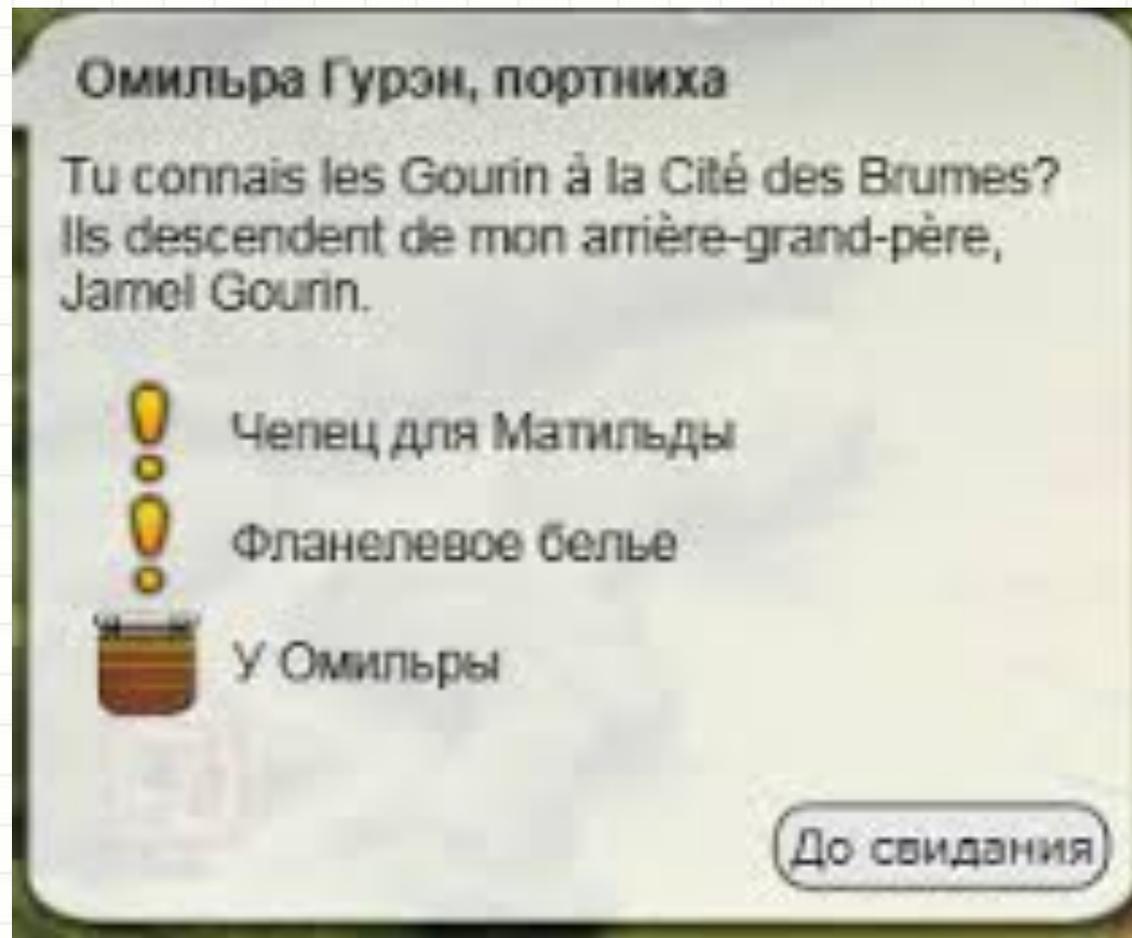
Тестирование локализации

Локализация – процесс адаптации программного продукта к языку и культуре клиента. Данный процесс адаптации включает в себя:

- Перевод пользовательского интерфейса.
- Перевод документации.
- Контроль формата даты и времени.
- Внимание к денежным единицам.
- Внимание к правовым особенностям.
- Раскладка клавиатуры пользователя.
- Контроль символики и цветов.
- Толкование текста, символов, знаков.
- И т.д.

Тема: **ТЕСТИРОВАНИЕ**

Тестирование локализации и интернализации

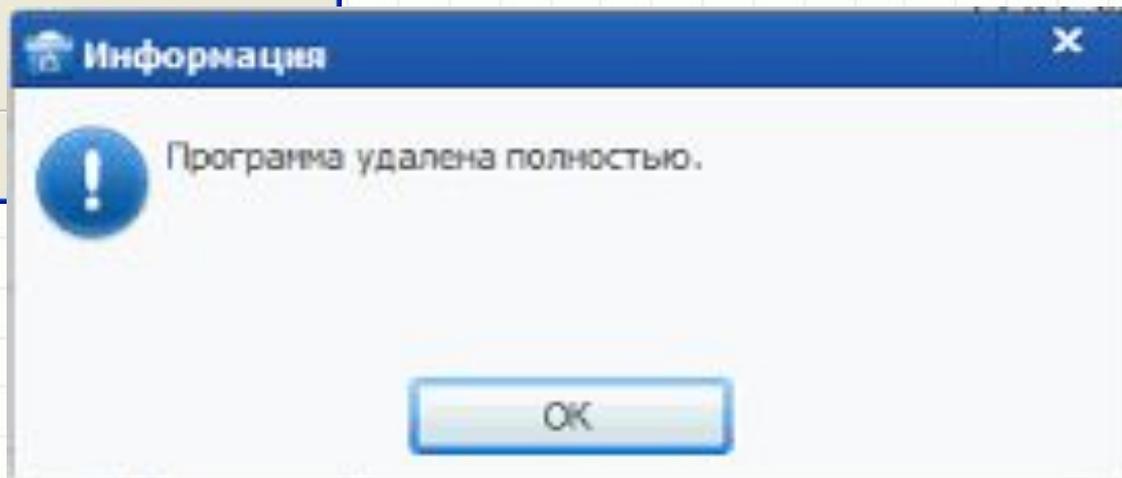
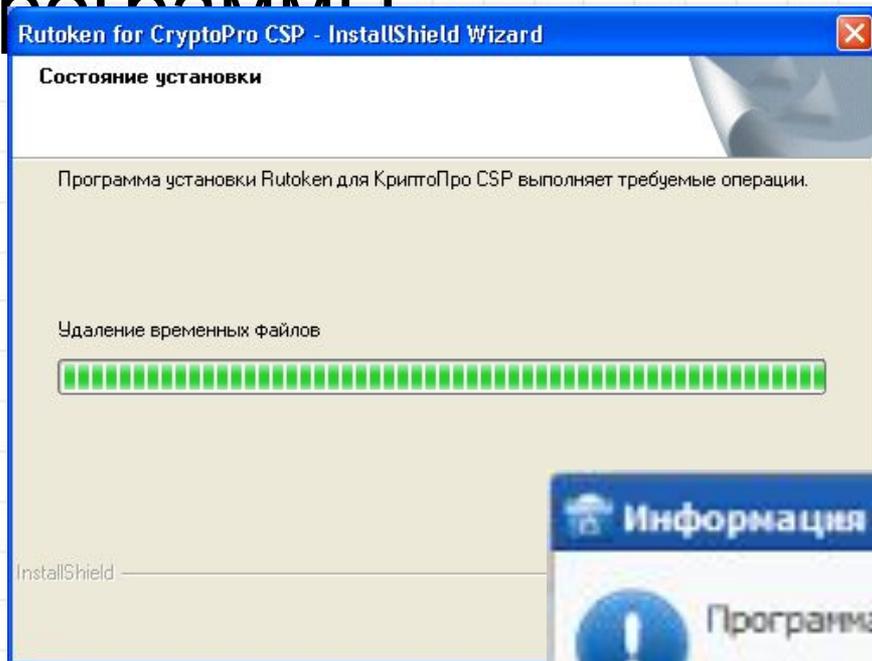


Тема: **ТЕСТИРОВАНИЕ**

Инсталляционное тестирование

Тестирование установки и удаления

ПРОГРАММЫ



Тема: **ТЕСТИРОВАНИЕ**

Конфигурационное тестирование

Разное оборудование (тип и количество процессоров, объем памяти, характеристики сети / сетевых адаптеров), разные программные средства (ОС, драйвера и библиотеки, стороннее ПО, влияющее на работу приложения и т.д.), разные браузеры, разное разрешение экрана и т.д



Configuration Testing

Тема: **ТЕСТИРОВАНИЕ**

Тестирование на отказ и восстановление

- Проверяет способность продукта противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками программного обеспечения, отказами оборудования или проблемами связи (например, отказ сети).
- Целью данного вида тестирования является проверка систем восстановления (или дублирующих основной функционал систем), которые, в случае возникновения сбоев, обеспечат сохранность и целостность данных тестируемого продукта.

Тема: ТЕСТИРОВАНИЕ

Тестирование на отказ и восстановление

Пример сбоев, которые при тестировании эмулируются:

- Внезапный отказ электричества на компьютере
- Потеря связи с сетью
- Отказ носителей
- Наличия в системе неверных данных

Тема: **ТЕСТИРОВАНИЕ**

ТЕСТ АНАЛИЗ

Тест анализ – выбор что именно будет протестировано. Причем выбор производится таким образом, чтобы протестировать **все важное и не пропустить критичных ошибок.**



*К сожалению полное исчерпывающее тестирование невозможно
Приходится выбирать!*

Тема: **ТЕСТИРОВАНИЕ**

ДЕКОМПОЗИЦИЯ ФУНКЦИОНАЛЬНОСТИ

Для начала продукт необходимо **исследовать**.
Для этого функционал продукта **декомпозируют** (делят) на составные части и определяют **взаимосвязи** между этими составными частями.

Дает нам это:

- более глубокое понимание продукта (в том числе прояснение ранее неучтенных моментов)
- возможность более детально оценить трудозатраты на выполнение проекта

Тема: ТЕСТИРОВАНИЕ

ШАГИ ИССЛЕДОВАНИЯ ПРОДУКТА

- 1. Сбор входящей информации (требования, аналоги разрабатываемого продукта и т.д)**
- 2. Наглядное представление информации**
- 3. Поиск потерь**
 - с помощью согласования с заказчиком и в проектной команде**
 - с помощью других методов исследования**

Тема: ТЕСТИРОВАНИЕ

Продукт для примера - <http://sitechco.ru/>

The screenshot displays the Sitechco web application interface. At the top, there is a navigation bar with the Sitechco logo and menu items: "На главную", "Проекты", "Настройки", and "Справка". The user's name "Наталья" and a power button are on the right. Below the navigation bar is a dashboard with six main menu items: "Дашборд", "Чек-листы", "Отчеты", "Требования", "Тест-план", and "Задачи". A "СВЕРНУТЬ МЕНЮ" button is located at the bottom right of the dashboard.

Below the dashboard, the breadcrumb "Танцы с отчетом > Чек-листы" is shown. A toolbar contains several icons: a play button, a plus sign in a square, a folder with a plus sign, a pencil, a trash can, a document, and a funnel. A dropdown menu on the right shows "Танцы с отчетом".

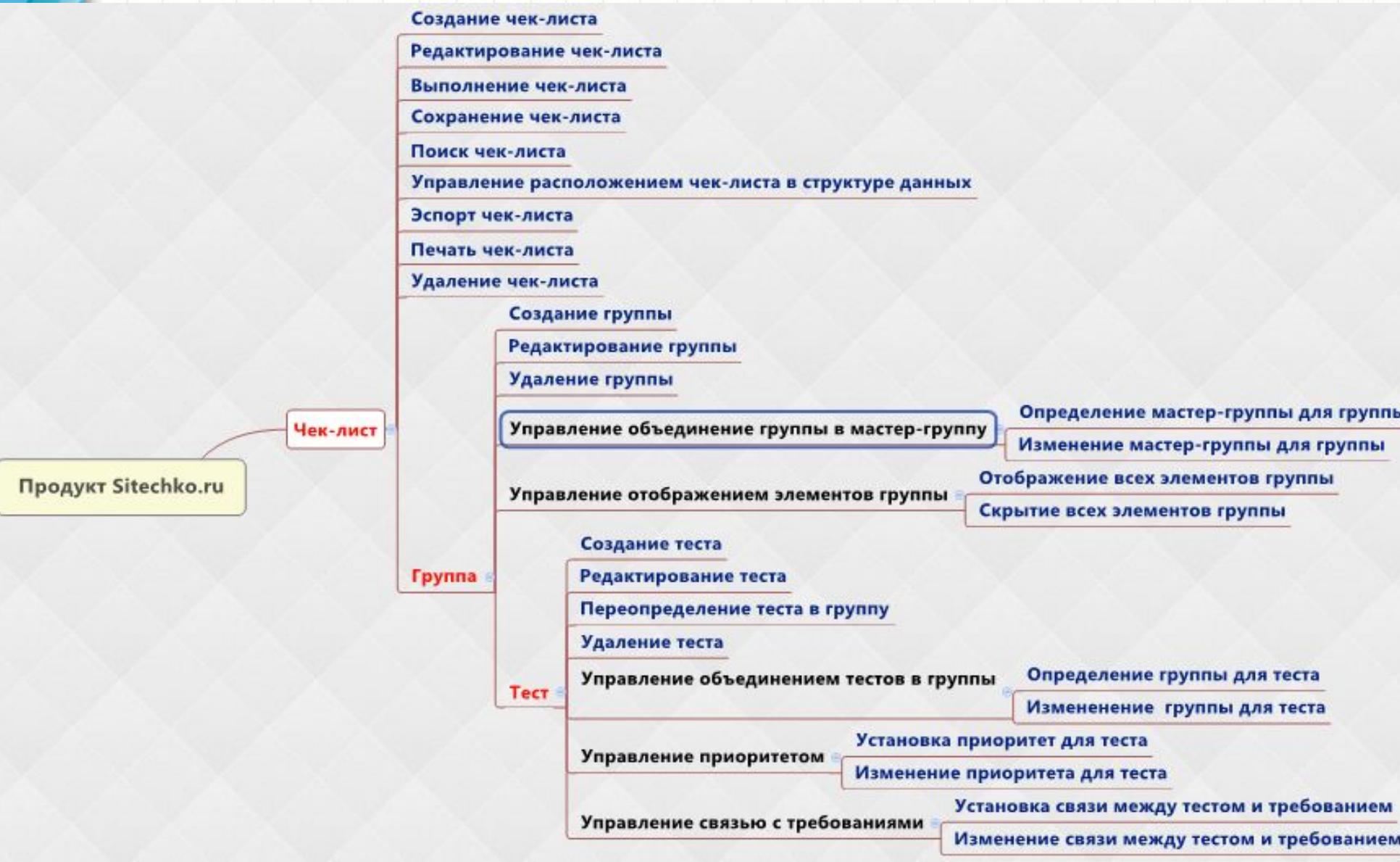
АББР.	НАЗВАНИЕ	ОПИСАНИЕ	ГОТОВНОСТЬ
<input type="checkbox"/>	Чек-листы	Чит-листы	ВСЕГО 0 ТЕСТОВ

У ВАС НЕТ НИ ОДНОГО ЧЕК-ЛИСТА

«Лаборатория качества» © 2010 Simply Test Check Cooperate Текущая версия: 140703

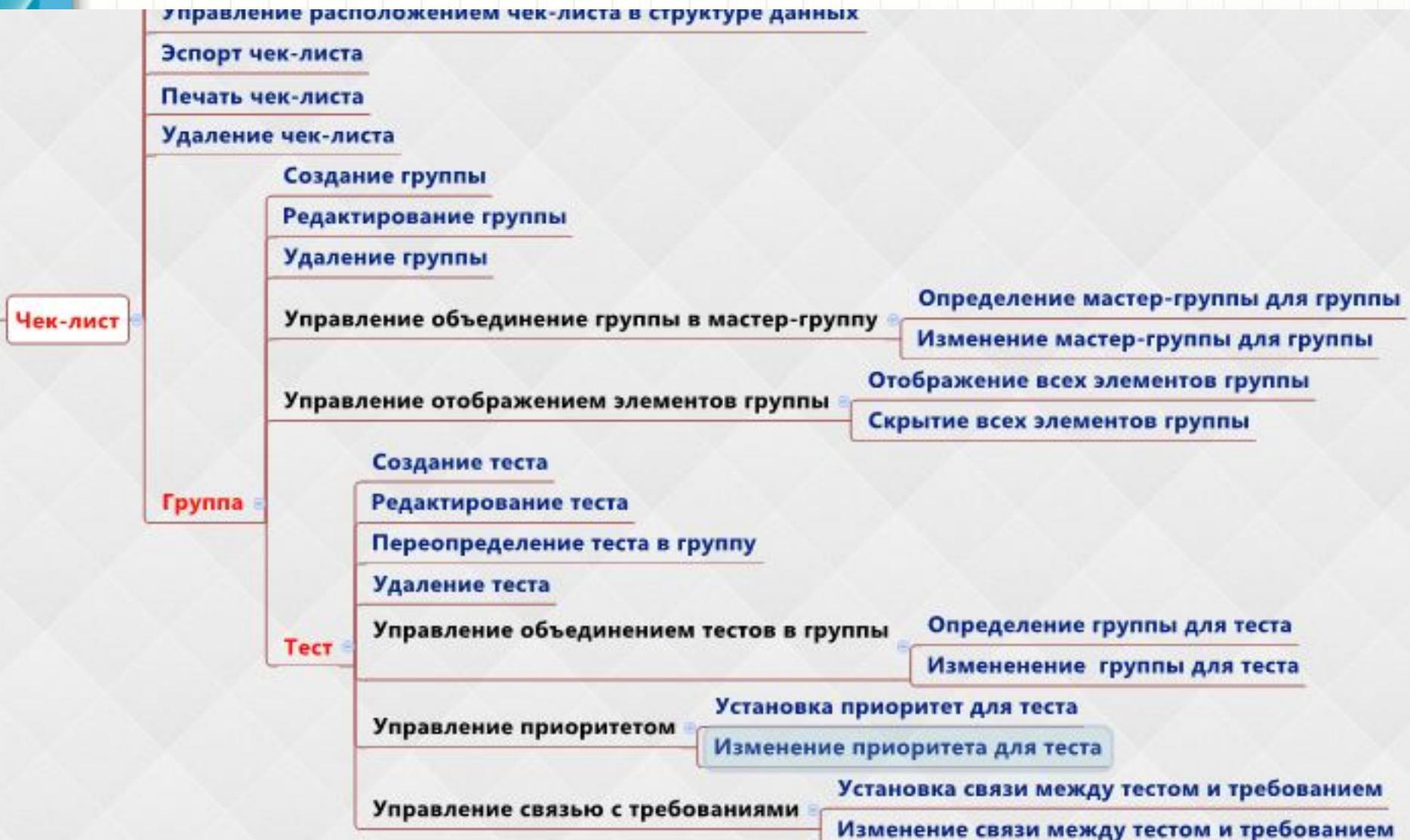
Тема: ТЕСТИРОВАНИЕ

Mind-Map: Объекты и действия



Тема: ТЕСТИРОВАНИЕ

Mind-Map: Объекты и действия



Тема: ТЕСТИРОВАНИЕ

Правила построения Mind-Map Объекты- Действия

**1. На первом уровне декомпозиции –
“Какой есть объект?”.**

Например: чек-лист, группа, тест

**2. На втором уровне декомпозиции -
“Что с этим объектом можно
сделать?”**

Тема: **ТЕСТИРОВАНИЕ**

Исследование объектов

Далее исследуют объекты. **Для каждого действия**, которое можно произвести с объектом выписывают **параметры**, влияющие на выполнение данного действия и оценивают **значения**, которые данные параметры могут принимать.

При анализе значений, которые могут принимать параметры, используются техники тест-дизайна:

- **Классы эквивалентности**
- **Анализ граничных значений**

Тема: **ТЕСТИРОВАНИЕ**

Исследование объектов

**Сначала
выписываются
параметры,
которые есть в
пользовательском
интерфейсе
(пользовательские
параметры) и
исследуются их
возможные
значения.**

Создание нового чек-листа

НАЗВАНИЕ * РЕГУЛЯРНОСТЬ ПРОХОЖДЕНИЯ
ОСТАЛОСЬ: 100 СИМВОЛОВ Не указано ▼

АББРЕВИАТУРА * ТИП ЧЕК-ЛИСТА
Проверки и результаты ▼

ОПИСАНИЕ

ИМПОРТ ТЕСТОВ
Вставьте тесты, которые Вы хотите добавить к чек-листу. Для этого Вы можете просто скопировать нужные строки из таблицы Excel или Google Docs в это поле. Каждая строка импортируемого текста должна содержать информацию по одному тесту. Текст для колонок «Проверка» и «Ожидаемый»

ТРУДОЗАТРАТНОСТЬ ТЕГИ
 ИСПОЛЬЗОВАТЬ ТЕГИ ДЛЯ ТЕСТОВ

Действие “Создание чек-листа”, параметр “Название”

Название	Длина		0					
			1					
			100					
			20 (default)					
			101					
	Состав	Цифры						
		Буквы	Регистр			Строчные (default)		
						Заглавные		
			Алфавит/Письмо				Классический латинский алфавит (default)	
							Модификации (диакритические знаки, лигатуры) латинского алфавита	
							Кириллица	
							Греческий алфавит	
							Армянское письмо	
						Еврейский алфавит		
					Арабское письмо			
					Армянское письмо			
			Грузинское письмо					
			Семейство индийских писем					
				Китайское письмо (иероглифы)				
	Различные кодировки							
	Спецсимволы							
	Уникальность		Да (default)					
			Нет					
	Обязательность		Да (default)					
			Нет					

Действие “Создание чек-листа”, параметр “Аббревиатура”

Аббревиатура	Длина		0			
			1			
			2 (default)			
			4			
			5			
	Состав	Цифры			Регистр	Строчные (default) Заглавные
			Буквы	Алфавит/Письмо	Различные кодировки	Классический латинский алфавит (default)
		Модификации (диакритические знаки, лигатуры) латинского алфавита				
		Кириллица				
		Греческий алфавит				
		Армянское письмо				
		Еврейский алфавит				
		Арабское письмо				
		Армянское письмо				
Грузинское письмо						
Семейство индийских писем						
Китайское письмо (иероглифы)						
Спецсимволы						
Уникальность	Да (default)					
	Нет					
Обязательность	Да (default)					
	Нет					

Аналогично для описания, трудозатратности, тип, описания и т.д определяются значения.

При этом негативные проверки (т.е недопустимые значения) выделены **красным, допустимые значения – **зеленым**, значения по умолчанию – **черным**.**



См. файл

Microsoft
Excel

Исследование связей анализируемого объекта с другими параметрами

Далее анализируем связи между анализируемым объектом и всеми его параметрами и всеми остальными объектами.

Например, а как задача влияет на чек-лист.

- **Можно ли создать чек-лист при создании новой задачи?**
- **Можно ли создать чек-лист, если статус задачи Готова?**
- **Можно ли создать чек-лист, если статус задачи Отменено?**
- **Можно ли создать чек-лист, если статус задачи В процессе?**
- **Можно ли создать чек-лист при редактировании задачи?**
- **Можно ли создать новый чек-лист при редактировании существующей задачи, в случае если к ней уже подключен чек-лист?**
- **Есть ли соответствие между трудозатратами за задачи и трудозатратами чек-листа?**

Исследование связей параметров внутри объекта.

Далее анализируем связи между параметрами внутри объекта.

Как определяются значения параметров?

- *Классы эквивалентности*
входные параметры, которые приводят к **одинаковому поведению программы**, мы будем считать эквивалентными.

Классы эквивалентности

Алгоритм использования техники такой:

- **Определить классы эквивалентности**
- **Выбрать одного представителя от каждого класса.**
- **Выполняем тест со значением параметра от каждого класса эквивалентности.**

Классы эквивалентности

Можно разбивать тесты на классы эквивалентности по разным принципам.

Например, если мы тестируем поле ввода, которое принимает максимум 5 символов, то мы можем выбрать разные принципы разбиения на классы эквивалентности:

- По количеству символов
- По типу символов (цифры, буквы, спец символы)

Исследование объектов

Классы эквивалентности - пример

Подсчет комиссии при отмене бронирования авиабилетов:



- За 5 суток до вылета комиссия составляет 0%
- Меньше 5 суток, но больше 24 часов – 50%
- Меньше 24 часов, но до вылета – 75%
- После вылета – 100%

Исследование объектов

Классы эквивалентности - пример

Определим классы эквивалентности (для каждого теста из этих классов мы ожидаем получить одинаковый результат):

1 класс: время до вылета > 5 суток

2 класс: 24 часа $<$ время до вылета < 5 суток

3 класс: 0 часов $<$ время до вылета < 24 часа

4 класс: время до вылета < 0 часов (вылет уже состоялся)

Исследование объектов

Классы эквивалентности - пример

Выберем любого представителя от каждого класса.

время до вылета = 10 суток (тест из 1-го класса)

время до вылета = 3 суток (тест из 2-го класса)

время до вылета = 12 часов (тест из 3-го класса)

время до вылета = -30 мин (тест из 4-го класса)

Исследование объектов

Классы эквивалентности - пример

Проведем тесты

- **Отменим бронь за 10 суток до вылета и проверим, что комиссия составила 0%.**
- **Отменим бронь за 3 суток до вылета и проверим, что комиссия составила 50%.**
- **Отменим бронь за 12 часов до вылета и проверим, что комиссия составила 75%.**
- **Отменим бронь через 30 мин после вылета и проверим, что комиссия составила 100%.**

Исследование объектов

Анализ граничных значений

Это **проверка ошибок** на границах классов эквивалентности.

Считается, что с **граничными значениями** связаны серьезные риски, так как даже если эквивалентные классы найдены правильно, то граничные значения могут быть ошибочно отнесены к другому классу.

Исследование объектов

Анализ граничных значений

Алгоритм использования техники анализа граничных значений:

- 1. Выделить классы эквивалентности.**
- 2. определить граничные значения этих классов.**
- 3. понять, к какому классу будет относиться каждая граница.**
- 4. Для каждой границы нам нужно провести тесты по проверке значения до границы, на границе, и сразу после границы.**

Исследование объектов

Анализ граничных значений - пример



Выделим классы

эквивалентности:

время до вылета > 5 суток

24 часа \leq время до вылета ≤ 5 суток

0 часов $<$ время до вылета < 24 часа

время до вылета ≤ 0 часов (вылет уже состоялся)

Исследование объектов

Анализ граничных значений - пример

Определим границы:

5 суток

24 часа

0 часов

**Определим, к какому классу
относятся границы:**

5 суток – к 2-му классу

24 часа – к 2-му классу

0 часов – к 4-му классу

Исследование объектов

Анализ граничных значений - пример

Протестируем значения на границах, до и после них:

- **Отменим бронь за 5 суток + 1 секунду до вылета (или просто постараемся выполнить бронь как можно ближе к границе, но слева от нее) и проверим, что комиссия равна 0%.**
- **Отменим бронь ровно за 5 суток до вылета и проверим, что комиссия равна 50%.**
- **Отменим бронь за 5 суток – 1 секунду до вылета и проверим, что комиссия равна 50%.**
- **Отменим бронь за 24 часа + 1 секунду до вылета и проверим, что комиссия равна 50%.**

Исследование объектов

Анализ граничных значений - пример

- **Отменим бронь ровно за 24 часа до вылета и проверим, что комиссия равна 50% Отменим бронь за 24 часа - 1 секунду до вылета и проверим, что комиссия равна 75%.**
- **Отменим бронь за 1 секунду до вылета и проверим, что комиссия равна 75%.**
- **Отменим бронь ровно во время вылета и проверим, что комиссия равна 100%.**
- **Отменим бронь спустя 1 секунду после вылета и проверим, что комиссия равна 100%.**

Тема: Тестирование требований к ПО

Критериям качества требований:

Полнота

Непротиворечивость (consistency).

Корректность (correctness).

Недвусмысленность (unambiguousness).

Проверяемость (verifiability).

Модифицируемость (modifiability).

Прослеживаемость (traceability).

Проранжированность по (ranked for...)

- важности (importance).
- стабильности (stability).
- срочности (priority).



Тема: Анализ требований заказчика к

ПО

Пример требования:

Тр4: После набора номера пользователь должен слышать короткие гудки (символизирующие о том, что идет дозвон)

Пример результатов его проверки

№	Требования	Критерий	Детальное описания замечания
1	Тр4	Корректность	После набора номера пользователь должен слышать долгие гудки (символизирующие о том, что идет дозвон) согласно {указать источник информации}, а не короткие

Тема: Анализ требований заказчика к ПО

Пример требования:

*Тр 192 (из раздела функциональности спец-кнопок):
когда активен режим «Mute», телефон не должен издавать никаких звуков;*

Тр245 (из раздела интерфейса): когда пользователь снимает трубку, телефон должен издавать тоновые гудки.

Пример результатов его проверки

№	Требования	Критерий	Детальное описания замечания
2	Тр192, Тр 245	Противоречивость	Противоречие между требованиями Тр192 и Тр245 в случае активного режима Mute и снятой трубки.

Тема: Анализ требований заказчика к ПО

Пример требования:

*Тр 192 (из раздела функциональности спец-кнопок):
когда активен режим «Mute», телефон не должен издавать никаких звуков;*

Тр245 (из раздела интерфейса): когда пользователь снимает трубку, телефон должен издавать тоновые гудки.

Пример результатов его проверки

№	Требования	Критерий	Детальное описания замечания
2	Тр192, Тр 245	Противоречивость	Противоречие между требованиями Тр192 и Тр245 в случае активного режима Mute и снятой трубки.

Тема: Анализ требований заказчика к ПО

Пример требования:

Тр 248: информация на дисплее телефона должна отображаться в понятном пользователю виде

Пример результатов его проверки

№	Требования	Критерий	Детальное описания замечания
2	Тр 248	Проверяемость	Не специфицировано, что такое понятный вид для пользователя. Т.о определить критерий прохождения теста/ожидаемые результаты не предоставляется возможным.

Структура работы:

Красным выделены обязательные пункты

I. Техническое задание

1. ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование системы и ее условное обозначение

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ

2.1 Назначение системы

2.2 Цели создания системы

3. ТРЕБОВАНИЯ К СЦЕНАРИЯМ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

3.1 Перечень сценариев использования системы

3.2 Пользователи системы

3.3 Описание сценариев использования системы (хотя бы 2 самых сложных)

4. ТРЕБОВАНИЯ К ДАННЫМ

4.1 Требования к составу и форме предоставления входной информации

4.2 Требования к составу и форме предоставления выходной информации

•

5. ТРЕБОВАНИЯ К ОТЧЕТНОСТИ

5.1 Перечень отчетов системы

5.2 Описание отчетов системы (хотя бы одного)

6. ТРЕБОВАНИЯ К ФУНКЦИЯМ СИСТЕМЫ

6.1 Перечень функций

6.2 Требование к функции 1

6.3 Требование к функции 2

6.4 Требование к функции 3

....

7. ТРЕБОВАНИЕ К ИНТЕРФЕЙСАМ

7.1 Требования к внутренним интерфейсам

7.2 Требования к внешним интерфейсам

7.2.1 Пользовательские интерфейсы

7.2.2 Интерфейсы со смежными системами

8. ТРЕБОВАНИЯ К УДОБСТВУ ИСПОЛЬЗОВАНИЯ

8.1 Требования к пользовательскому интерфейсу

8.2 Требование к наличию отчетов

8.3 Требование к пользовательской документации

8.4 Требование к средствам обучения

9. ТРЕБОВАНИЯ К НАДЕЖНОСТИ

9.1 Требование к отказоустойчивости

9.2 Требования к сохранности данных

10. ТРЕБОВАНИЯ К ПРОИЗВОДИТЕЛЬНОСТИ

10.1 Требования к времени отклика

10.2 Требования к нагрузке

11. ТРЕБОВАНИЯ К ЗАЩИТЕ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

12. ТРЕБОВАНИЯ СОВМЕСТИМОСТИ

13. ТРЕБОВАНИЕ К ВИДАМ ОБЕСПЕЧЕНИЯ

13.1 Требование к программному обеспечению

13.2 Требование к аппаратному обеспечению

II. Конкретизация сценариев использования системы

III Изучение системы

а) Mind Map или таблица: иерархии объектов в системе (все объекты)

б) Mind Map или таблица: объект - возможное действие над ним (все объекты)

в) Mind Map или таблица: действие - атрибуты действия (для всех действий над одним объектом)

г) Ассоциации между объектами (диаграмма)

д) Исходная диаграмма классов

IV Процесс ICONIX

Диаграмма прецедентов (проверяется соответствие диаграммы вариантов использования из ТЗ).

Диаграмма пригодности

Диаграмма последовательности (совмещенная с вариантами использования) – для двух вариантов использования